UNIVERSIDAD CARLOS III DE MADRID

BACHELOR'S DEGREE IN COMPUTER SCIENCE & ENGINEERING

Facing the Hard Truth: Showing Users What Mobile Apps Can Learn About Them From the Location Data They Collect

---

Author
Adolfo Santiago Mouge

Supervisor
Marco Gramaglia

June, 2018

Contents

1 Introduction **1**

2 State of the Art **3**
2.1 Location trackers in Android . . . 3
2.2 Software . . . 3
2.2.1 Lumen . . . 3
2.2.2 *Location Privacy Droid* (LPDroid) . . . 4
2.2.3 *Facebook Data Valuation Tool* (FDVT) . . . 4
2.3 Previous work with location data . . . 5

3 Project Management **6**
3.1 Objectives . . . 6
3.2 Development environment . . . 6
3.2.1 Android Studio and Kotlin . . . 7
3.2.2 Third-Party Libraries . . . 7
3.2.3 Version Control and Bug tracking . . . 7
3.2.4 Trello . . . 8
3.2.5 Hardware . . . 8
3.3 Methodology . . . 8
3.4 Development phases . . . 8
3.5 Use Cases . . . 9
3.6 Requirements . . . 11
3.6.1 Functional Requirements . . . 11
3.6.2 Non-Functional Requirements . . . 14
3.6.3 Use Cases traceability . . . 16

4 Project Development **17**
4.1 Software Design . . . 17
4.2 Application Architecture . . . 18
4.2.1 Location Service . . . 18
4.2.2 Database Manager . . . 21
4.2.3 Map Visualization . . . 22
4.2.4 Preference Manager . . . 23
4.2.5 Determining the *staypoints* . . . 24

5 Testing **28**
5.1 Application testing . . . 28

5.2 User validation of the data . . . 29
5.2.1 User information . . . 29
5.2.2 Points gathered . . . 29
5.3 Staypoints running the KDD algorithm . . . 30
5.3.1 Tests for the dataset provided by *Anon #3* . . . 30
5.3.2 Tests for the dataset provided by *Anon #4* . . . 32
5.3.3 Tests for the dataset provided by *Anon #6* . . . 34
5.4 Conclusion on the testing . . . 36
5.4.1 User feedback . . . 37

6 Legal Framework and Socieconomic Context **39**
6.1 Ethical Consideration . . . 39
6.2 Legal Framework . . . 39
6.2.1 *Protección de Datos de Carácter Personal* . . . 39
6.2.2 Foreign countries . . . 41
6.2.3 Free and Open Source Software . . . 41
6.3 Socioeconomic Context . . . 43
6.3.1 Budget . . . 43

7 Conclusions **45**

8 Future work **46**

References **51**

List of Tables

3.1 Use Case Template . . . 9
3.2 Use Case UC-01. Gathering location . . . 10
3.3 Use Case UC-02. Store location data . . . 10
3.4 Use Case UC-03. Visualize data stored . . . 10
3.5 Use Case UC-04. Determine the staypoints of the user . . . 10
3.6 Use Case UC-05. Permissions in applications . . . 10
3.7 Requirement Template . . . 11
3.8 Functional Requirement FR-01. Location data gathering . . . 11
3.9 Functional Requirement FR-02 . . . 11
3.10 Functional Requirement FR-03. Minimum time interval location . . . 12
3.11 Functional Requirement FR-04 Minimum distance location . . . 12
3.12 Functional Requirement FR-05. Location data gathered notification . . . 12
3.13 Functional Requirement FR-06. Location data storage . . . 12
3.14 Functional Requirement FR-07. Export stored data . . . 12
3.15 Functional Requirement FR-08. Import stored data . . . 13
3.16 Functional Requirement FR-09. Delete stored data . . . 13
3.17 Functional Requirement FR-10. Map visualization . . . 13
3.18 Functional Requirement FR-11. Stored data visualization . . . 13
3.19 Functional Requirement FR-12. Determine staypoints . . . 13
3.20 Functional Requirement FR-13. Notifying determined staypoints . . . 14
3.21 Functional Requirement FR-14. Visualizing staypoints . . . 14
3.22 Functional Requirement FR-15. Configure parameters . . . 14
3.23 Functional Requirement FR-16. Location-enabled permission applications . . . 14
3.24 Non-Functional Requirement NFR-01. Location information in a map . . . 14
3.25 Non-Functional Requirement NFR-02. Location latitude . . . 15
3.26 Non-Functional Requirement NFR-03. Location longitude . . . 15
3.27 Non-Functional Requirement NFR-04. Location time span . . . 15
3.28 Non-Functional Requirement NFR-05. Location altitude . . . 15
3.29 Non-Functional Requirement NFR-06. Location accuracy . . . 15
3.30 Non-Functional Requirement NFR-07. Minimum Version of Android . . . 15
3.31 Requirement - Use Case Traceability . . . 16

4.1 Database Scheme . . . 21

5.1 Points gathered by user . . . 30
5.2 Test for *Anon #3* . . . 30
5.3 Test for *Anon #4* . . . 33

5.4 Test for *Anon #6* 35

6.1 Personnel total costs 43
6.2 Equipment costs 43
6.3 Project costs 44

List of Figures

3.1 Gantt chart for the project 9

4.1 MVC Pattern architecture 17
4.2 Application Architecture 18
4.3 Location Providers 19
4.4 A-GPS Scheme 19
4.5 Location Providers with `PASSIVE_PROVIDER` 20
4.6 Map Visualization Example. a) Left image with a cluster, b) Right image with clusters 23
4.7 `Knowledge Discovery in Databases (KDD)` Algorithm execution 25
4.8 `KDD` Algorithm execution with Accuracy 26
4.9 Map Visualization Example. a) Left image with normal data, b) Right image with staypoints 27

5.1 Location points by *Anon #3* 31
5.2 Staypoints calculated for *Anon #3*: Test 1, Test 2, Test 3, Test 4, Test 5 32
5.3 Location points by *Anon #4* 33
5.4 Staypoints calculated for *Anon #4*: Test 3 and Test 4 34
5.5 Location points by *Anon #6* 35
5.6 Staypoints calculated for *Anon #6*: Test 1, Test 2, Test 3, Test 4, Test 5 36
5.7 Responses about the number of location points 37
5.8 Responses about the knowledge extraction from the data 38
5.9 Responses about the resource consumption 38

8.1 New Application Architecture with VPN 46

Listings

4.1 Location Data Format 20
4.2 KDD Original algorithm 24
4.3 KDD Modified algorithm 25

Abstract

We live in a world where data is food to companies. Users install more applications each day in their smartphones without taking into account all the data gathering it is done to offer new or improved "services" or to show personalized content based on that data.

One kind of data companies gather is the location data. With this data, companies are capable of personalize Advertisement (ads) shown to the user with fine-grained precision, and also making possible to follow the user knowing what places the user visited.

Users need to know how much location data has been queried from their smartphone and be able to look at it in any instant.

From that necessity of awareness *The PAPAYA tool (**P**rivacy leackages from **a**p**p** tr**a**jector**y** d**a**ta (PAPAYA))* has been born. It is a tool that records any location update in the smartphone and allows to look at the data within the application. It is been also implemented a *KDD* algorithm to know the most stationary places the user has been to.

Moreover a research on the legal and socio-economic implications of this project is included.

Keywords: data, location, privacy, awareness

Resumen

Vivimos en un mundo donde los datos son el alimento de las empresas. Cada vez los usuarios instalan más aplicaciones en sus terminales móviles sin tener en cuenta los datos que son recolectados para ofrecer y mejorar "servicios" y mostrar contenido personalizado en base a esos datos.

Uno de los datos más importante para las empresas es la localización de los usuarios. Con esos datos, las empresas son capaces de personalizar la publicidad que muestra a los usuarios con una granularidad muy baja, y además, realizar un seguimiento de los lugares donde el usuario ha estado.

Los usuarios deben conocer cuántos datos de localización se han recolectado en su terminal, así como poder visualizar dichos datos en el momento.

De esta necesidad de concienciación ha nacido *The PAPAYA tool*, una herramienta que recoge todas las peticiones de localización realizadas en un terminal móvil y permite visualizar dichos datos dentro de la aplicación. Además, se realiza la implementación de un algoritmo KDD, usado para conocer las zonas donde el usuario ha pasado más tiempo.

Asímismo, se provee de un estudio del marco legislativo bajo el que los datos son recogidos, así como un estudio socio-económico de este proyecto.

Palabras claves: datos, localización, privacidad, concienciación

Acknowledgements

Gracias a mis familia por todo su apoyo durante mi vida y educación y por su inmensa paciencia.

Gracias a todos mis amigos durante los años de Universidad, han sido un gran soporte. Las noches en vela haciendo trabajos no habrían sido las mismas sin echarnos unas risas juntos.

Gracias a Marco, mi supervisor, por la oportunidad que me ofreció y su paciencia y ayuda durante el desarrollo del proyecto, ha sido un gran guía.

Gracias.

1 Introduction

Business is driven now by user data. Whenever an application is installed in a smartphone, it needs a number of permissions to access to the phone and to offer services and functionality. Sometimes permissions are asked, sometimes they must be accepted without choice. But that comes with a price. In the now called "Orwellian society", the privacy users enjoyed the time before no longer exists as such.

The applications users have in their smartphones are usually propietary, meaning the user does not know what it is really happening within the application and with their data. They gather a considerable amount of data to process and to show personalized content based on that data. The most known example of this comportment is showing ads based on what the information the user generates within the application [1] (Section *How do we use this information?*).

Those applications are usually free of charge. Users do not usually read the Privacy Policy (PP) of the applications and the main reason is because of the complexity and length of the text. At other times those terms are opaque and it is not possible to see how the data is stored and processed. Hence the principal problem with the definition of "free". For example, Twitter for Android gather what applications the user installed in its phone [2] (Section 2.4, *Log Data*). The user will not know this behavior if the PP of the application is not read.

Usually every application asks to make an account in order to use its services. In that case your data is protected under their security systems, so only the user has access to it. Some services include the option to activate the 2-Factor Authentication (2FA) to add an extra step protecting the data. But the data stored in the servers is still shareable with third-party partners to offer personalized content.

Businesses often collect several types of data, one of which is location data. With location data it is possible to track where the user has been and set up a set of preferences and personalized content based on those locations. But another problem that data gathering presents is the discovering of places users have been like hospitals, residence of the user, shopping centers the user was, among other places.

It is mandatory to add that the Internet Service Provider (ISP) that provides services to users like making calls, sending Short Message Service (SMS) and connecting to Internet through mobile data is tracking the user too. An ISP tracks an user by using the area where mobile phones are connected to get signal. Moreover, *LocationSmart*, a firm known as a location aggregator, buys information from ISP carriers to get the location data. A recent new shows it was used without consent from the users [3].

The PAPAYA tool (PAPAYA) is a project to raise awareness about privacy between users who are constantly using platforms which gather data, in particular location data. In an era where data drives business it is important to recognize when users are giving a high amount of data. It is not possible to know if businesses really comply with their PP but it is a step forward into that direction.

With this idea in mind this project intents to gather location data, store it internally (which means not being in a server) and applying an KDD algorithm to show the most common places the user was in a map.

This document is divided in eight principal chapters. First the *State of the Art* with the previous work to this project is exposed. Second the *Project Management* is explained with the objectives of the project, the software used for the project development and the timeline. After that the development of the architecture is described and how the data is gathered. Consequently to the development the testing and results section expose the data. The next stage shows the socio-economic study along with the legal framework and the budget. Finally, the document concludes with the conclusions and the future work for this project.

This project is motivated by the idea to know how many location data is gathered and provide the user with a comprehensive list of those applications. It is quite common users give the "Location" permission to an application and forget about it.

The PAPAYA tool (PAPAYA) is a project funded by the Data Transparency Labs [4] as a 2017 grantee.

2 State of the Art

Before getting into the development process it is mandatory to discuss some work in this section in order to put some insight in gathering location data.

Although some companies perform anonymization techniques over the data, it is not the scope of this project so it is not being summed up any kind of technique to perform anonymization or related to that process.

2.1 Location trackers in Android

Android, as an open ecosystem, makes possible to develop applications of almost any kind. In the Google Play Store an user can look for applications which allow the user to store its location data. Those applications, however, do not allow the user to export their data and make an analysis of it without being root users in their phones.

As an example, it is possible to mention *Traccar* [5], which is an application that allows the users to store their location data in a server. It is not possible to perform any analysis of the data in the phone but it may be possible to see the data and the user trajectory in the server application, which is in a website. However, the data does not appear in a cluster or something but just points.

The main problem is data is just from one application, *Traccar* itself. So it is not a valid approach for this project. Also it is server-dependent, which this project does not have in its scope or as a long-term part to develop.

2.2 Software

There are a few works directly related with this project which are being discussed in the following sections.

2.2.1 Lumen

Lumen is part of the *ICSI Haystack Project* [6] and a 2016 grantee of the Data Transparency Labs (DTL) [4].

This project is a monitor in real-time for data leaks from the installed applications. It analyzes the mobile traffic and helps to identify data leaks which can be considered as privacy leaks. All the data gathered by Lumen is also used for research purposes.

However, the project is closed source, which is not a good way to get users to trust the application. Also the data, even used for research purposes, it is not downloadable as an individual, which means the user cannot know how much data *Lumen* has stored.

2.2.2 *Location Privacy Droid* (LPDroid)

LPDroid [7] is an extension of *TaintDroid* [8], a real-time data and privacy monitoring on smartphones developed for the Android platform.

The initial work of this project was an extension to the Android platform that analyzes in real-time the private data through third-party applications using dynamic taint analysis techniques, which is based on inspecting the executed source code. This extension is implemented at low-level, meaning it implies modifying the operating system.

Later, *TaintDroid* was modified to make *LPDroid*, an extension of the original project which main goal is to protect the location data of the users when an application request for a location update, modifying that location data with anonymization techniques and blocking the network connection for that application if this data is detected being leaked.

Although this approach will be valid for this project, a main problem is presented. Most of the users which use Android as their main operating system in their smartphones are not prone to modify the terminal in any way, from rooting the phone to even changing the stock rom for a custom one.

For this point of view, the possibility of modifying Android to make this project more accurate is better, but the user base will be smaller than using an application installed in the phone.

2.2.3 *Facebook Data Valuation Tool* (FDVT)

The *Facebook Data Valuation Tool* (FDVT) is a 2015 grantee of the Data Transparency Labs (DTL) [4].

This tool analyzes the Facebook account data of the user in real-time and shows the economic value of that data. The user has to provide the location, gender, age, birthday and relationship status.

However, this tool only provides some insight of the Facebook information and has to be used as an add-on to a browser (Mozilla Firefox, Google Chrome or Opera at this moment). Also it does not provide any kind of value speaking of location data.

2.3 Previous work with location data

Previous to this project it has been a few projects which deserve to be mentioned.

In the first work the authors used a dataset previously tagged with geolocation data from Twitter, and then asked the people who were participants to tag that data with the location they think it belongs, also using different data visualization techniques [9]. It shows that the functional locations of user activities can be inferred with high precision and low density datasets. However this project is done using data from Twitter which needs to contain the geolocation component.

The next work shows that the authors compared sources of location data from Android smartphones using forensic tools [10]. In this article the authors extracted data from internal databases of the applications and the phone itself and compared them with the data of a German network operator. This shows that the data generated from the phone is more accurate than the generated by the network. However, this project also needs the collaboration from a network operator and internal data only extractable with root permissions, which is not a part of this project.

The last work, which this project uses part of the content, is the capturing and analyzing data of people in the real world with a Global Positioning System (GPS) device [11]. Then, with data fixed in a distance and time threshold, determining the staypoint(s) of the users and make a cluster of those points. Later, the data is analyzed to show how knowledge is extracted from the analysis of the data. Participants of this article also had to provide some insight of the analyzed data in order to extract a conclusion. This work is interesting as shows how patterns are valueless if it is more frequent than others. The problem with this data is that the data is gathered by GPS phones and receivers, not day-to-day smartphones, which means it is high precise and not anyone has one at home.

3 Project Management

In this chapter it is going to be described how the project has been developed, including the objectives driving the project, the methodology used, phases of time and all the software used. After that the use cases and requirements are explained to collect the parts which form the application.

3.1 Objectives

All the data gathered for the previous work described in the chapter (section 2) is for academic use or business purposes, and in both cases the final users do not have access to the data. Moreover, the projects mentioned are closed source and the datasets are not individual but in community so, as an individual user, it is not possible to visualize all the data.

The conditions under the experiments were executed in the academic work can differ from the real world completely, and promising results shown in experiments cannot relate with normal conditions, meaning behaving otherwise.

PAPAYA is a project that can be used for academic work and for real users to gather data, process it and view it. The goals of this project are:

- **Determine how much points are gathered:** Too much applications installed in an Android smartphone are requesting location updates through GPS (Assisted GPS (A-GPS) if using the network too), so it is important to know how much location points are stored.
- **Determine the staypoint quantity:** It is important to know the most quantity of staypoints possible because it defines areas where the user spent more time than others.
- **Awareness of privacy:** At the the of this project the main idea is to ***raise awareness about the privacy issues*** for the users, because it is important to know how much data is harvested while using applications in the phone. With this goal it is in mind to reach a high user base.

3.2 Development environment

The development environment used for this project will be exposed in this section.

3.2.1 Android Studio and Kotlin

Android Studio was the first choice for Android development. While there are other Integrated Development Environment (IDE) to develop applications for Android, Android Studio provides all the native tools developed by Google for the Android platform, including the Android Debug Bridge (ADB), the debugger, the profiling tools, the code linter (although it is Java, Android has custom libraries in its Application Proggramming Interface (API)), among others.

The language of preference was Kotlin [12]. Kotlin offers a integration with the Java Virtual Machine (JMV) and it is interoperable with Java, which means it is possible to run Java code inside a Kotlin class, and vice versa. Also it has characteristics Java does not such a null-safety (aimed to eliminate the null references from code), companion objects, smart casts, string templates, among others (Kotlin website [12], Learn, FAQ section, *Comparison to Java*).

3.2.2 Third-Party Libraries

Some third-party libraries were necessary for the development of the application:

- `OSMDroid` [13] is the library for using the service of OpenStreetMap [14], a map service provider. It allows to use the application without the Google Play Services, which contains the Google Maps API, reaching a high base of interested users in the application.
- `OSMBonusPack` [15] is the library for complementing the functionality of the OSMDroid library. It adds clustering, routes and directions, points of interests, among others.
- `SQLiteImporterExporter` [16] is the library for exporting and importing the data from the database. It allows a fast way to export and import the dataset from and to the device and storing in a way that the user can see the content.

All the libraries used in this application are compatible with the license used for the project as stated in the section 6.2.3.

3.2.3 Version Control and Bug tracking

A Version Control System (VCS) is needed for the management of the source code and the bugs in the application. Between Git [17], Mercurial [18] and SVN [19], Git was the chosen one.

Git is a free and open source distributed version control system. Its open philosophy allows the project to be compatible with the idea of the free availability of the source code, distribution and modification. It is non-linear, which means some developers can get into the code without disrupting the other developers, and merge the changes into one single place, called the `master` branch, where the latest stable version is. This is another brilliant

feature of Git, the *branching*, which allows to partition the workflow and then merge it into the `master` branch.

GitHub [20] was the platform used to store the source code and the bug tracking. This allows to share the code and improve the reviewing process. Also provides features such rollbacks and branching to not mix up fixings or new features in the main code.

3.2.4 Trello

Trello [21] is a platform to organize tasks in form of cards inside boards. It follows the Kanban [22] methodology. It was used to manage all the tasks involved in the project.

3.2.5 Hardware

The computer used in this project was a mid-2015 MacBook Pro 13" Retina with 16GB of RAM and a SSD of 256GB running macOS v10.11.6 “Sierra”.

Two phones were used for Android deployment and testing: LG Nexus 4 (with the Google Play Store Services) and LG Nexus 5 (with a clean rom), both running Android Nougat 7.1.2.

3.3 Methodology

The methodology used for this project is the *Incremental model*. This model implements the Waterfall model incrementally.

This model was chosen because the project, although the objectives and the parts were defined previously, required a testing and debug phase for each component implemented. The order in the phases was important, but when a new component of the application was integrated it was necessary to debug and modify the actual source adding or fixing features. At the end the application was finished to test with a closed betatester group.

3.4 Development phases

In this section is presented the phases of the project and the *Gantt* chart reflecting the timeline of the project.

The project started looking for how Android manages the location updates inside the phone (without root).

After how the location updates works was made clear the process of developing a service which gathers that location started. The data it was collected contained a few fields unusable so they were discarded. Data is stored in a internal database with a custom engine to handle all the update operations.

Once data was stored it was implemented the visualization of the data to see the location points gathered by the service.

Finally the KDD algorithm was implemented to start determining the staypoints when the Universal Serial Bus (USB) is connected.

The following *Gantt* chart shows the different phases within a time span.

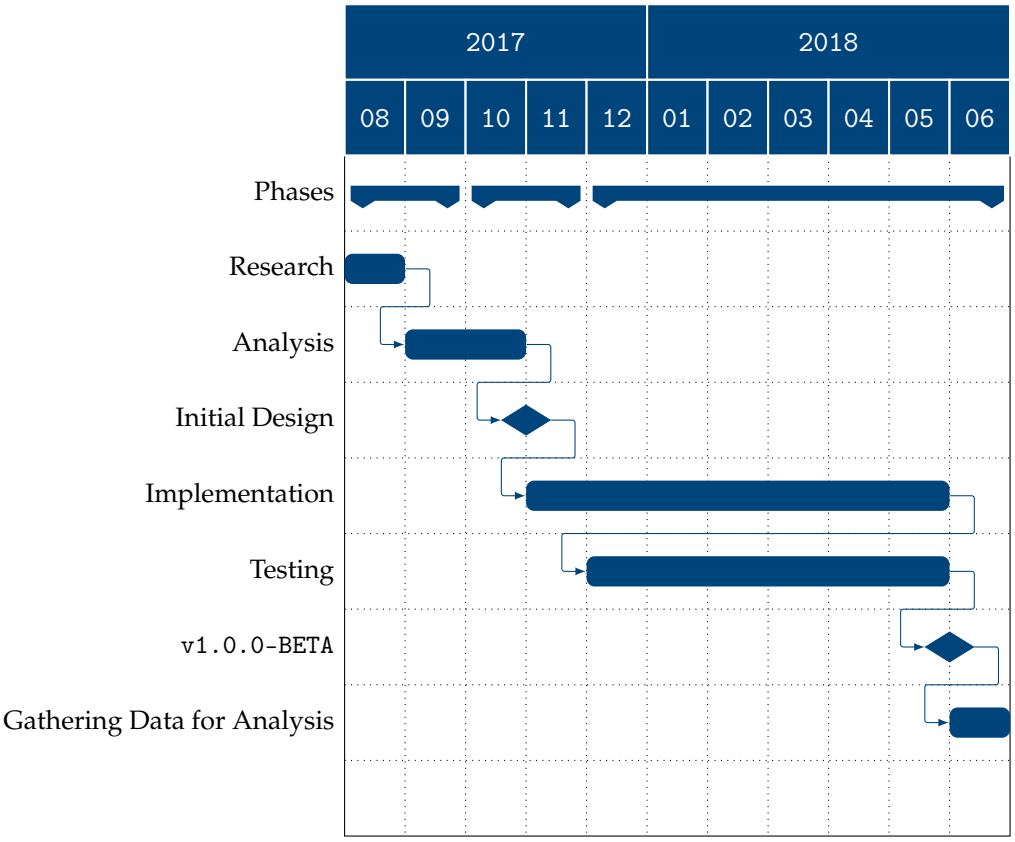

FIGURE 3.1: Gantt chart for the project

3.5 Use Cases

In this section the *Use Cases* for the application will be defined. An *Use Case* defines a specific way of using the system by an actor.

ID	UC-XX
Title	
Actor	
Preconditions	
Description	

TABLE 3.1: Use Case Template

The use cases specified in this section will be defined by the template described in the table 3.1. Each use case will receive an unique identifier described by `UC-XX`, where `XX` is a

double digit number. Later this identifier will be used for the traceability of requirements.

ID	UC-01
Title	**Gathering location**
Actor	Application
Preconditions	Applications have to access the location of the device
Description	Gather the location data at the same moment third-party applications make a location request to the system

TABLE 3.2: Use Case UC-01. Gathering location

ID	UC-02
Title	**Store location data**
Actor	Application
Preconditions	A location request must be done by at least a third-party application
Description	Data gathered will be stored in the device

TABLE 3.3: Use Case UC-02. Store location data

ID	UC-03
Title	**Visualize data stored**
Actor	User
Preconditions	Location data must be available in the internal storage
Description	Location data stored will be viewable through a map

TABLE 3.4: Use Case UC-03. Visualize data stored

ID	UC-04
Title	**Determine the staypoints of the user**
Actor	Application
Preconditions	Location data must be available in the internal storage
Description	The application will be determine staypoints of the user

TABLE 3.5: Use Case UC-04. Determine the staypoints of the user

ID	UC-05
Title	**Permissions in applications**
Actor	User
Preconditions	Applications with the permission location enabled must be installed
Description	The user will be able to access to an application list which have the location permission enabled

TABLE 3.6: Use Case UC-05. Permissions in applications

3.6 Requirements

After defining the Use Cases (section 3.5) for this project it is mandatory to define the requirements. Requirements are divided in two categories:

- **Functional:** A functional requirement is any requirement which describes what the system should do.
- **Non-Functional:** A non-functional requirement is any requirement which describes a constraint to the system.

Both kind of requirements will be described the template described in the table 3.7.

For both requirements it will available an unique `ID`, `FR-XX` and `NFR-Y-XX`, with two digit numbers `XX` (which can or cannot be the same).

ID	FR-XX || NFR-XX
Title	
Description	
Priority	
Use Case(s)	

TABLE 3.7: Requirement Template

Priority can be *high*, *medium* or *low* depending of how important is the requirement to get the system working.

The "Use Case(s)" row will not be in the non-functional requirements because they are not related with the user but how the system performs to achieve that function.

3.6.1 Functional Requirements

ID	FR-01
Title	**Location data gathering**
Description	The application shall be able to gather location data
Priority	High
Use Case(s)	UC-01

TABLE 3.8: Functional Requirement FR-01. Location data gathering

ID	FR-02
Title	**Location data from applications**
Description	The application shall gather location data only when a third-party application requests the location of the phone
Priority	High
Use Case(s)	UC-01

TABLE 3.9: Functional Requirement FR-02

ID	FR-03
Title	**Minimum time interval location**
Description	The application shall allow to the user to configure the minimum time interval between location updates
Priority	Medium
Use Case(s)	UC-01

TABLE 3.10: Functional Requirement FR-03. Minimum time interval location

ID	FR-04
Title	**Minimum time distance location**
Description	The application shall allow to the user to configure the minimum distance interval between location updates
Priority	Medium
Use Case(s)	UC-01

TABLE 3.11: Functional Requirement FR-04 Minimum distance location

ID	FR-05
Title	**Location data gathered notification**
Description	The application shall allow the user to enable a notification when location data is gathered
Priority	Low
Use Case(s)	UC-01

TABLE 3.12: Functional Requirement FR-05. Location data gathered notification

ID	FR-06
Title	**Location data storage**
Description	The application shall store the gathered location data in the internal storage of the device
Priority	High
Use Case(s)	UC-02

TABLE 3.13: Functional Requirement FR-06. Location data storage

ID	FR-07
Title	**Export stored data**
Description	The application shall allow the user to export stored data
Priority	Medium
Use Case(s)	UC-02

TABLE 3.14: Functional Requirement FR-07. Export stored data

ID	FR-08
Title	**Import stored data**
Description	The application shall allow the user to import stored data
Priority	Medium
Use Case(s)	UC-02

TABLE 3.15: Functional Requirement FR-08. Import stored data

ID	FR-09
Title	**Delete stored data**
Description	The application shall allow the user to delete stored data
Priority	Medium
Use Case(s)	UC-02

TABLE 3.16: Functional Requirement FR-09. Delete stored data

ID	FR-10
Title	**Map visualization**
Description	The application shall provide a Graphical User Interface (GUI) with a map
Priority	High
Use Case(s)	UC-03

TABLE 3.17: Functional Requirement FR-10. Map visualization

ID	FR-11
Title	**Stored data visualization**
Description	The application shall display stored location data
Priority	High
Use Case(s)	UC-03

TABLE 3.18: Functional Requirement FR-11. Stored data visualization

ID	FR-12
Title	**Determine staypoints**
Description	The application shall determine the staypoints of the user
Priority	High
Use Case(s)	UC-04

TABLE 3.19: Functional Requirement FR-12. Determine staypoints

ID	FR-13
Title	**Notifying determined staypoints**
Description	The application shall notify when staypoints are determined
Priority	High
Use Case(s)	UC-04

TABLE 3.20: Functional Requirement FR-13. Notifying determined staypoints

ID	FR-14
Title	**Visualizing staypoints**
Description	The application shall allow to visualize the staypoints
Priority	High
Use Case(s)	UC-04

TABLE 3.21: Functional Requirement FR-14. Visualizing staypoints

ID	FR-15
Title	**Configure parameters**
Description	The application shall allow the user to configure the necessary parameters to determine the staypoints
Priority	Medium
Use Case(s)	UC-04

TABLE 3.22: Functional Requirement FR-15. Configure parameters

ID	FR-16
Title	**Location-enabled permission applications**
Description	The application shall access to the installed applications list with location permission enabled
Priority	High
Use Case(s)	UC-05

TABLE 3.23: Functional Requirement FR-16. Location-enabled permission applications

3.6.2 Non-Functional Requirements

ID	NFR-01
Title	**Location information in a map**
Description	The application shall show information about location data in a map
Priority	High

TABLE 3.24: Non-Functional Requirement NFR-01. Location information in a map

ID	NFR-02
Title	**Location latitude**
Description	The location information shall have latitude
Priority	High

TABLE 3.25: Non-Functional Requirement NFR-02. Location latitude

ID	NFR-03
Title	**Location longitude**
Description	The location information shall have longitude
Priority	High

TABLE 3.26: Non-Functional Requirement NFR-03. Location longitude

ID	NFR-04
Title	**Location time span**
Description	The location information shall have time span
Priority	High

TABLE 3.27: Non-Functional Requirement NFR-04. Location time span

ID	NFR-05
Title	**Location accuracy**
Description	The location information shall have accuracy
Priority	High

TABLE 3.28: Non-Functional Requirement NFR-05. Location altitude

ID	NFR-06
Title	**Location provider**
Description	The location information shall have the provider
Priority	High

TABLE 3.29: Non-Functional Requirement NFR-06. Location accuracy

ID	NFR-07
Title	**Minimum Version of Android**
Description	The application shall run in an Android phone with the Android 4.1 *Jellybean* (API 16)
Priority	High

TABLE 3.30: Non-Functional Requirement NFR-07. Minimum Version of Android

3.6.3 Use Cases traceability

To understand the relation between use cases and functional requirements it is required to do a visual representation of the relation in the traceability matrix. This is necessary given that requirements must satisfy the use cases. Non-Functional requirements are not included since those are constraints to the functional requirements.

The representation is done in the table 3.31.

	UC-01	UC-02	UC-03	UC-04	UC-05
FR-01	✓				
FR-02	✓				
FR-03	✓				
FR-04	✓				
FR-05	✓				
FR-06		✓			
FR-07		✓			
FR-08		✓			
FR-09		✓			
FR-10			✓		
FR-11			✓		
FR-12				✓	
FR-13				✓	
FR-14				✓	
FR-15				✓	
FR-16					✓

TABLE 3.31: Requirement - Use Case Traceability

4 Project Development

In this chapter the design and implementation of *the PAPAYA tool* is explained along with the decisions in the process.

4.1 Software Design

The Model-View-Controller (MVC) pattern was chosen for the design of the architecture .

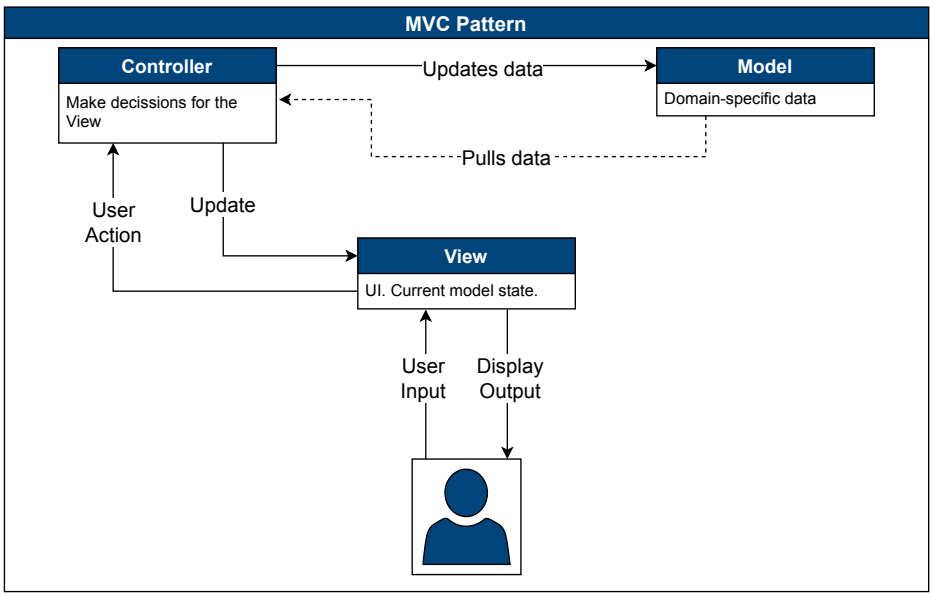

FIGURE 4.1: MVC Pattern architecture

This pattern allows to keep the parts of the application separated and communicate each other through the *Controller*. It makes the application modular which means it is easier to add new components and communicate each one of the through a common *Controller*.

The three parts of the scheme are described as follows:

- **View:** The *View* is with what the user views, the interface. It allows to show the stored data and configure the application to meet the requirements of the user for the available elements.
- **Controller:** The *Controller* is what handle all the requests made by the input of the user. It will update the data of the model and serve the information for the user.
- **Model:** The *Model* is the scheme of the data of the application. It manages the data of the application.

4.2 Application Architecture

In this section it will be explained how the application was implemented through the MVC pattern described in the section 4.1.

The application is *rootless*, which means it does not need any root permission to work. This makes possible to reach a higher quantity of users than using a root design. As stated in the *Objectives* (section 3.1), one of the goals of this project is to reach an user base as high as possible.

The description of the architecture will comprehend the location providers, how the data is stored and updated, the GUI (with all the components), and the background services and broadcast receiver needed to gather and process data, respectively.

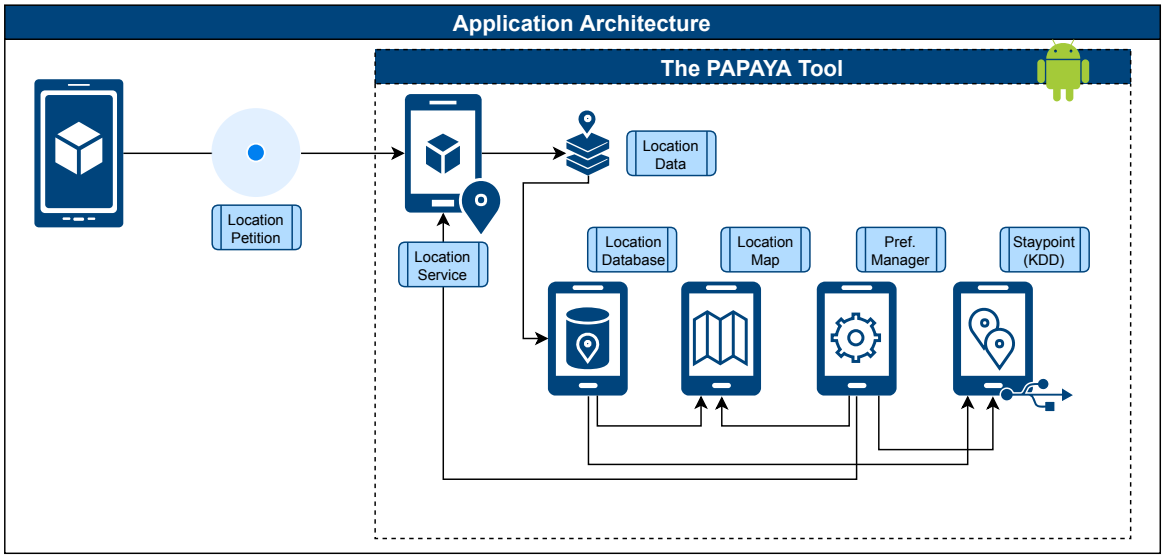

FIGURE 4.2: Application Architecture

The application, as seen in the figure 4.2, is composed by five main components. A description of each of them is provided in the following sections.

4.2.1 Location Service

This component of the application gathers the location updates made in the phone. It is composed of the `LocationManager` [23] and a background service [24] to keep executing without blocking the phone or the GUI.

There are two main location providers in the Android operating system: `GPS_PROVIDER` and `NETWORK_PROVIDER` [23] (Sections *GPS_PROVIDER* and *NETWORK_PROVIDER*). If the Android smartphone has the Google Play Services installed there is another location provider, the `FUSED` provider [25].

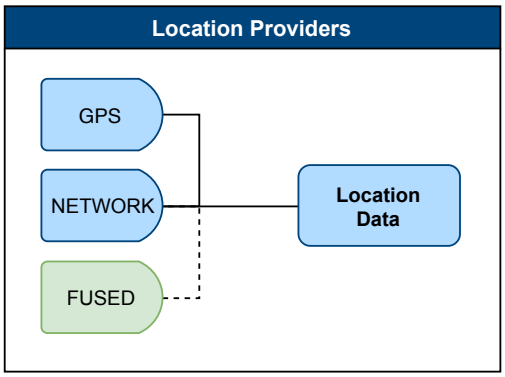

FIGURE 4.3: Location Providers

The Android operating system allows the user to use those providers simultaneously or separately:

- `GPS_PROVIDER` will only use the GPS in the phone to get a fixed location for the user. However, bad weather conditions make the GPS being less accurate.
- `NETWORK_PROVIDER` will use only the network connection to get a fixed location for the user. If the user is connected to the Wi-Fi the location is determined by the location of the router. If the user is connected to the mobile data the location is determined by the position of the antennas, giving the nearest tower location.
- `FUSED` combines GPS + Network, making the A-GPS, to improve the startup time of the GPS signal and get a location as soon as possible.

The system chooses the location provider automatically based on the accuracy and signal power to make location updates.

Using A-GPS makes possible to get information about orbital data for making the acquisition of satellites faster or calculating the position by the server using information from the GPS receiver [26].

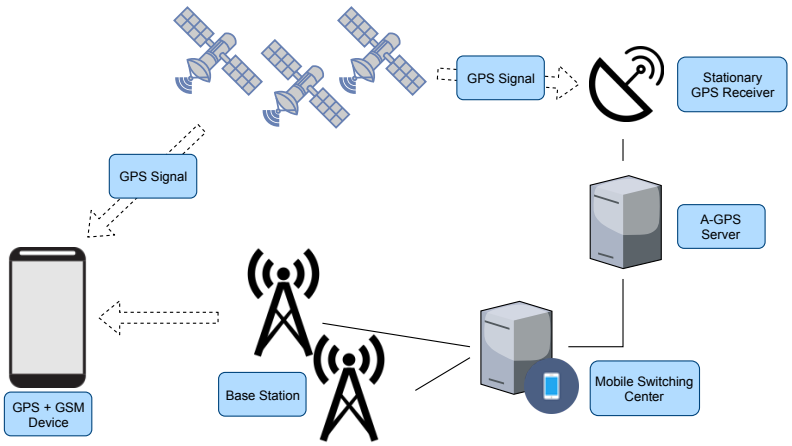

FIGURE 4.4: A-GPS Scheme

In the Android operating system itself exists a third location provider called `PASSIVE_PROVIDER` [23] (Section *PASSIVE_PROVIDER*).

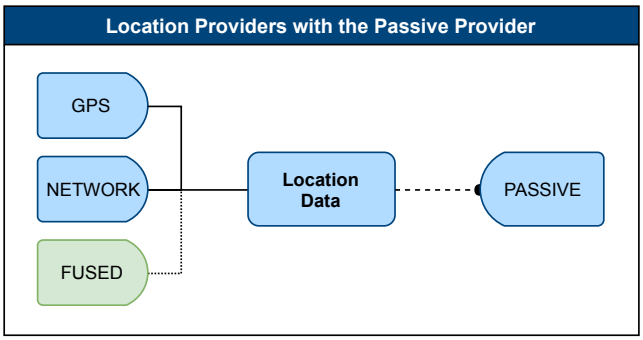

FIGURE 4.5: Location Providers with `PASSIVE_PROVIDER`

The `PASSIVE_PROVIDER` allows to receive passively location updates from other applications or services without requesting the location itself.

Note that the `FUSED` location provider is external to the Android operating system itself, which makes it the fourth of the whole ecosystem with Google. Users without the Google Play Services will not have this location provider available.

For being able to look for locations updates a background service was made [24]. This service handles the data gathering from the locations updates and passes the data to the *Database Manager* (section 4.2.2).

The data is received in the following form:

```
1 Location[PROVIDER LAT,LON,acc=ACC_VAL et=ET_TIME {Bundle[
     mParcelledData.dataSize=384]}]
```

LISTING 4.1: Location Data Format

A description of the values is provided as follows:

- `PROVIDER`: The provider of the location. The possible values are `gps`, `network` or `fused`.
- `LAT`: The `latitude` of the location.
- `LON`: The `longitude` of the location.
- `ACC_VAL`: The `accuracy` of the location (in GPS is a value less than using the Network provider).
- `ET_TIME`: The `timestamp` of the location (both UTC time and elapsed real-time since boot).
- `Bundle`: Data in the container for a message. It is not important information and is not stored.

This service does not start if the user did not give the consent to the application to use the location (permissions `ACCESS_COARSE_LOCATION` and `ACCESS_FINE_LOCATION` for the network and the GPS provider, respectively [23]).

The `LocationManager` needs to be set up with default values for the minimum time interval and minimum distance between location updates. Both of this values are set to 1000ms and

1m, respectively. This values are set to get the maximum quantity of location updates and reduce the battery consumption to the minimum possible. A value under 1000ms or 1m can cause a high battery consumption for the user. It is necessary to make a balance between data gathering and battery consumption.

4.2.2 Database Manager

The data gathered by the location service has to be stored in the phone. While it was stated the data will be stored internally in the phone.

Android provides the `SQLite` database engine [27] integrated into the operative system.

This allows to the application to store all the data internally and not using a third-party library outside the operating system or a third-party server (even one installed by the user).

A custom controller was developed for managing the updates in the database. It introduces a modular form to add methods that helps the application to add new queries into the manager.

When the location service receives an update, the data is passed into the database manager and the database is created if it does not exist (in the first start of the application). The data is stored, then, for being viewed or analyzed later.

The database follows the following scheme:

Column Name	Type	NOT NULL
ID	INTEGER, PK	✓
PROVIDER	TEXT	✓
LATITUDE	REAL	✓
LONGITUDE	REAL	✓
TIMESTAMP	DATETIME	✓
ACCURACY	REAL	✓

TABLE 4.1: Database Scheme

The `ID` is generated automatically each time a location update is received. `REAL` means a floating-point number. `DATETIME` is the format for dates of `SQLite`.

When the database is created it has access only by this application. For security and privacy reasons the database is not shareable internally with any other application.

Export and import the data
Users can export and import datasets gathered by this application. `SQLiteImporterExporter`, as mentioned in the section 3.2.2, is the library used.

It exports the database into the internal storage of the device within a file with extension `.db`. Later the user can import the same dataset again into the application.

4.2.3 Map Visualization

The application provides a visualization of the data in a map, presenting the information in an intuitive way for the users. The map shows the information in form of *location points*.

The data stored in the device, based in the scheme described in the table 4.1, contains the information for the visualization in a map (using the libraries `OSMDroid` and `OSMBonusPack` mentioned in 3.2.2).

When the location service gathers location updates and stores them in the database the user can visualize all the data in the map. As shown in the figure 4.6 *a)*, the location points the user can visualize are represented by a red dot. The image also shows a blue point with a number, called a *cluster point*.

As shown in the figure 4.6 *b)*, if there are a high number of points in a radius, the library `OSMBonusPack` provides a way to make a cluster of those points. This helps to show a total number of points in a giving zone of the map.

As the user accesses to the map the data is extracted from the database through the *Database Manager* (section 4.2.2). This dataset is time-fixed, meaning the user only extracts a subset of the complete database. By default is fixed at the last 24h. It is necessary to set the minimum time-fixed data extraction due to memory restrictions sharing data with a *Fragment* [28] in Android a high number of points can cause a force close to the application because of the high size of the set passed to the map.

When the user accesses to the map and points (with or without clusters) are visualized, the library downloads the tiles automatically and makes a cache in the internal memory of the phone, so if the user goes back to visualize the same points (or within the same area), the local cache is loaded. In the case of the user not connected to the Internet in that moment, the map does not provide a local OpenStreetMap provider since it will occupy a lot of space.

When the KDD algorithm is executed, the staypoints are visible in the map too, as shown in the figure 4.9.

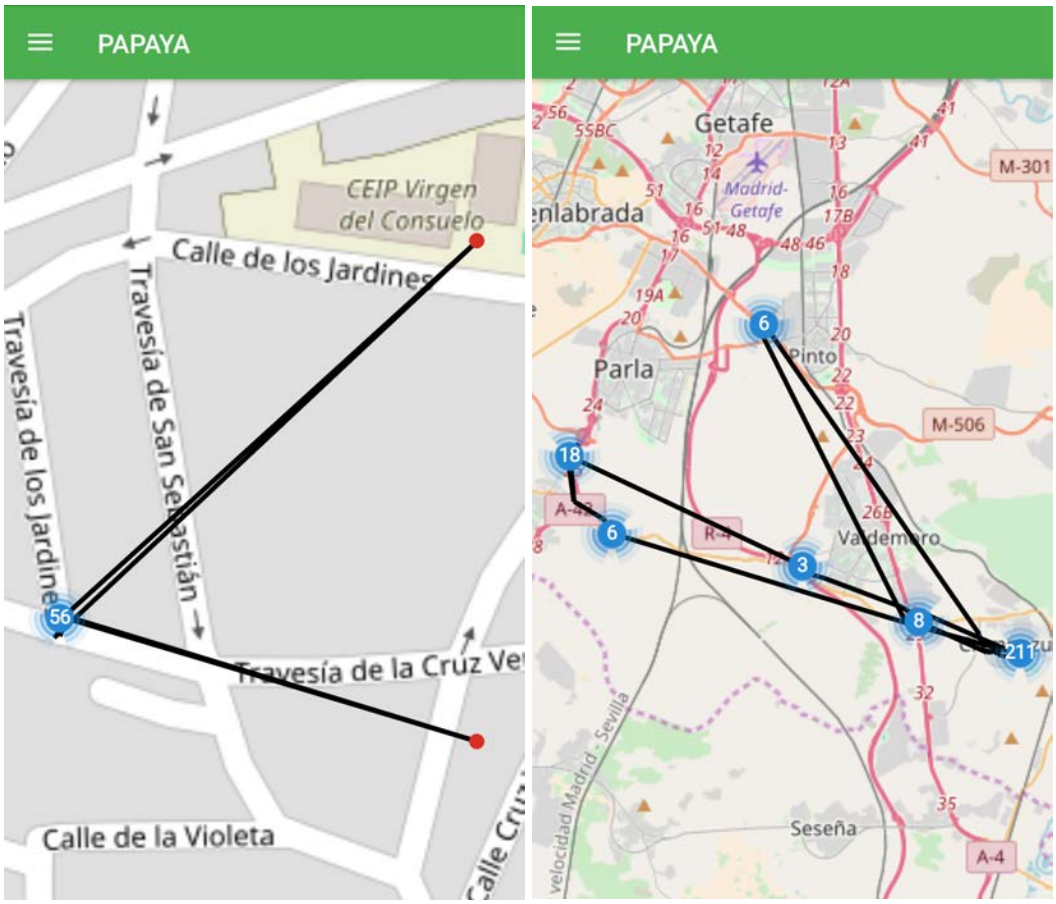

FIGURE 4.6: Map Visualization Example. a) Left image with a cluster, b) Right image with clusters

4.2.4 Preference Manager

Some of the values of the location service, the map, the KDD algorithm and the interaction with the database might be changed by the user for improving battery life or accuracy in order to get the staypoints.

The *Preference Manager* is a custom component of the application made for a easy access to the settings for the different components of the application.

As presented in the figure 4.2, this component is connected with the *Location Service* (section 4.2.1), the *Map Visualization* (section 4.2.3) and *the `KDD` algorithm* (section 4.2.5).

- **Settings for the *Location Service*:** It allows to change parameters in the `LocationManager` setup in order to configure the minimum time and distance between locations updates, and to enable a toast notification.
- **Settings for the *Map Visualization*:** It allows to enable visual components to the map such a multi-touch control or zoom control in screen.
- **Settings for the `KDD algorithm`:** It allows to change the parameters of the algorithm to make it more accurate.

- **Options for the *Database Manager*:** It allows to export, import and delete the data stored by the application.

4.2.5 Determining the *staypoints*

While the data stored seems to not have any meaning a priori, knowledge can be extracted from that data. The trajectory of the user can reveal patterns of behavior but the most concurrent places the user has been can reveal personal locations that, maybe, the user does not want to share with the service providers.

For this purpose this component of the application shows, in a more friendly way, the relation between the most concurrent places the user has been in order to show a good insight of the knowledge a business can have. As an example, if the user visits a shopping center after going from its home, and stays there for a period of 3 or more hours, it can be inferred that is the shopping center of preference for that user.

The algorithm used for determining the staypoints of the location data stored in the phone of the user was extracted from the paper mentioned in the *State of the Art* (section 2.3). This is the called the KDD algorithm in this document.

This algorithm executes when the user connects the phone to a charging source (such an USB port or charger). While the phone is charging the algorithm is executed. In the moment the USB is disconnected, the algorithm stops executing. Android provides the `BroadcastReceiver` which allows to execute actions based on broadcasts [29].

The original algorithm, extracted from [11], is reflected as pseudo-code in the listing 4.2.

```
1  Input:
2      GPS log P, distance threshold distThreh, time span threshold
           timeThreh
3  Output:
4      A set of StayPoints SP={S}
5  i:=0, pointNum=|P| (number of points in P)
6  while(i < pointNum)
7  {
8      j:=i+1, Token:=0
9      while(j < pointNum)
10     {
11         dist:=Distance(p_i, p_j)
12         if(dist > distThreh)
13         {
14             time:=p_j.T-p_i.T
15             if(time > timeThreh)
16             {
17                 S.coord:=ComputeMeanCoord({p_k : i<=k<=j})
18                 S.arvT:=p_i.T, S.levT:=p_j.T
19                 SP.insert(S)
```

```
20                  i:=j, Token:=1
21              }
22              break;
23          }
24          j:=j+1
25      }
26      if(Token!=1) i:=i+1
27  }
```

LISTING 4.2: KDD Original algorithm

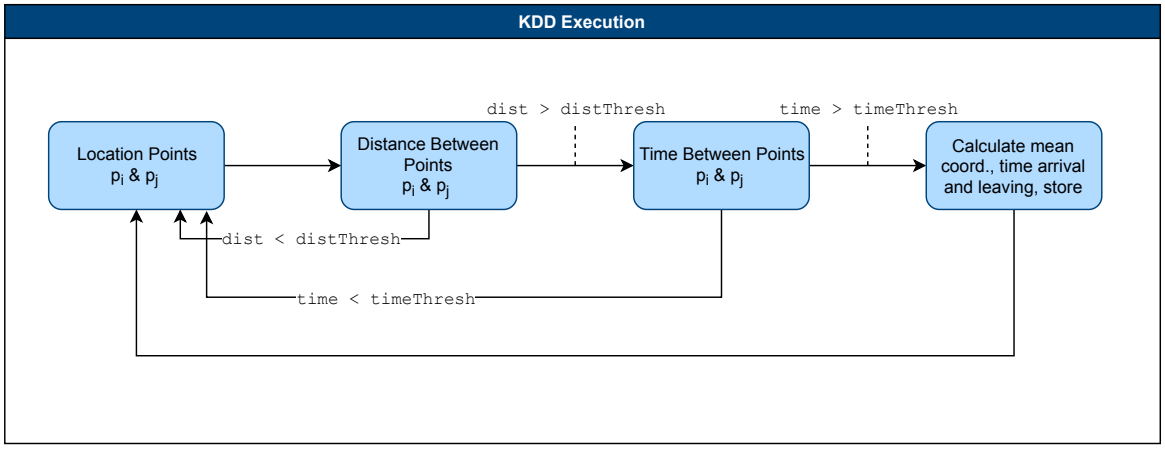

FIGURE 4.7: KDD Algorithm execution

When this algorithm was presented it was done by using GPS devices and GPS phones, only using the GPS technology and not taking into account the `network` use of the mobile phones. This presents the problem that accuracy in GPS is far precise than using the `network`, as the GPS triangulates the user with a high precision

After running some tests while implementing this component it was taken into account that the accuracy makes the algorithm to project bad staypoints or not enough precise in the map. For allowing the algorithm to use the accuracy in the calculations, the algorithm was modified by filtering all the points that do not pass an accuracy threshold.

The value of this accuracy threshold is 50m by default. It is a value based on the tests ran, as data shown it is a good value for getting staypoints.

```
1  Input:
2      GPS log P, distance threshold distThreh, time span threshold
           timeThreh, an accuracy threshold accThresh
3  Output:
4      A set of StayPoints SP={S}
5  i:=0, pointNum=|P| (number of points in P),
6
7  while(i < pointNum)
8  {
9      j:=i+1, Token:=0
```

```
10      while(j < pointNum)
11      {
12          dist:=Distance(p_i, p_j)
13          if(dist > distThreh)
14          {
15              time:=p_j.T-p_i.T
16              if(time > timeThreh)
17              {
18                  if(p_i.A <= accThresh && p_j.T <= accThresh)
19                  {
20                      S.coord:=ComputeMeanCoord({p_k : i<=k<=j})
21                      S.arvT:=p_i.T, S.levT:=p_j.T
22                      S.acc:=ComputeMeanAcc(p_i.A, p_j.A)
23                      SP.insert(S)
24                      i:=j, Token:=1
25                  }
26              }
27              break;
28          }
29          j:=j+1
30      }
31      if(Token!=1) i:=i+1
32  }
```

LISTING 4.3: KDD Modified algorithm

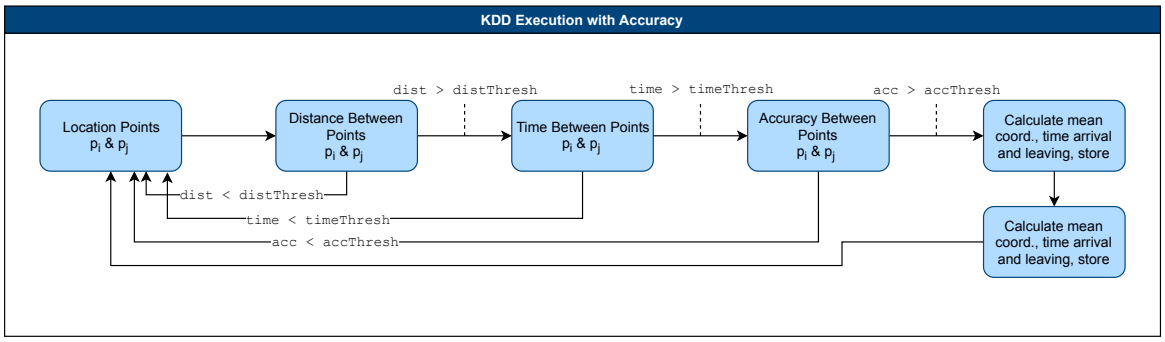

FIGURE 4.8: KDD Algorithm execution with Accuracy

The distance between points is calculated by the *Haversine formula* [30] (page 6). This formula is used to calculate the great-circle distance between two points, that means, the short distance over the surface of the earth.

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cdot\cos(\varphi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2\cdot\operatorname{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R\cdot c$$

Where φ means latitude, λ means longitude, and R is the radius of The Earth and its value is 6371.1370km. Δ means difference. Values of latitude and longitude are converted to radians in order to operate with the trigonometric functions.

Once the staypoints are calculated the user is allowed to see them in the map (section 4.2.3). In the next screenshots it is possible to see the location points gathered and the calculated staypoints by this algorithm. The staypoints are reflected as blue dots, instead the red dots of the normal data visualization.

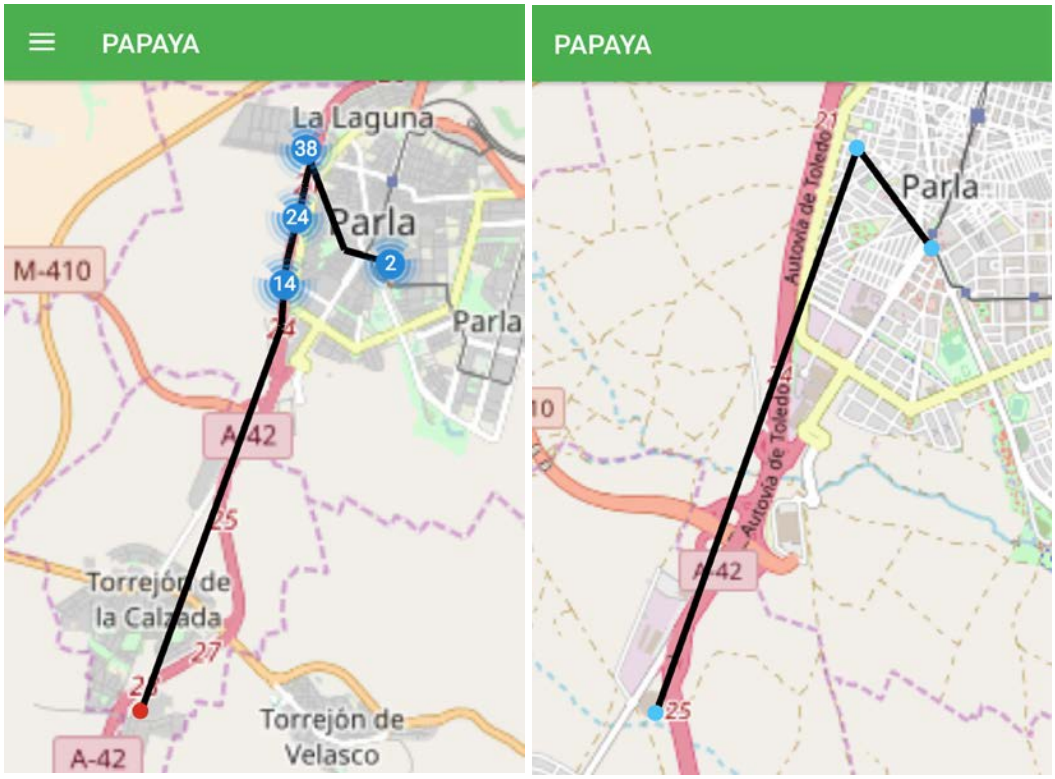

FIGURE 4.9: Map Visualization Example. a) Left image with normal data, b) Right image with staypoints

5 Testing

In this chapter it is going to be described the testing process of the application, involving the testing of each component and the real-world testing with a closed betatesters group. Also, a few executions of the KDD algorithm are exposed.

5.1 Application testing

This section describes the testing done to the application. The testing of the application had to be done in the development phases and it is not done by an automatic testing system.

Each component is tested the moment it is implemented, following the methodology described in the section 3.3, and fixing the problems encountered during the process. Doing the testing so it is possible to grant that the application works as intended.

The following points define the tests performed:

1. Testing the *Location Service*: The service defined in the section 4.2.1 is tested by using applications that make location updates inside and outside a building while the application is running.
2. Testing the *Database Manager*: The *Database Manager* defined in the section 4.2.2 is tested by performing updates in the data using the *Location Service* and visualizing the data within the map. Also from the *Preference Manager* with the *Deleting* option.

   For the *Export/Import* it is performed a test by exporting the database, erasing it and importing it, in which point the same data is visualized in the map.
3. Testing the *Map Visualization*: The *Map Visualization* defined in the section 4.2.3 is tested by loading the stored location data into the map and viewing if it is the data gathered by making location updates with the application running.
4. Testing the *Preference Manager*: The *Preference Manager* defined in the section 4.2.4 is tested by loading the default values and checking them within the application. Once the default values are loaded, those values are changed and tested if the new values are loaded correctly.

   If changes are made the new values are loaded when the application is started again.

5.2 User validation of the data

For a real-world testing the application was given to a closed group of betatesters. The users kept the application running during a week and the data was exported to analyze it.

From a legal point of view, users were informed about the purpose of the experiment and they gave their consent (specified in the section 6.1).

Data gathered for this thesis is not going to be use for bad purposes. This thesis pretends to raise awareness about the fact that, with a simple and open algorithm, it is possible to extract knowledge by the location updates generated. It is not possible to know what applications request location updates at this moment but it is planned as future work (chapter 8).

5.2.1 User information

The betatesters group was composed by eight users of Android.

- 2/8 users were running Android 8.1 Oreo. There is not any user running a custom version of the Android operating system.
- 6/8 users were running Android 7.1.2 Nougat. There is one user of this six using a custom Android operating system version.

The common applications between the users are the *Google* application, *Google Play Store*, *Google Maps*, *Telegram*, *WhatsApp* and *YouTube*.

Also several of them has the *LocationServices* provided by Qualcomm, *Waze*, some weather-related applications, *Amazon* application, *Wallapop*, among others. Note that the applications gathered for this study are those which are shown in the *Application List* in the application itself.

The seventh user does not have any kind of *Google*-related application except for *Google Maps*.

5.2.2 Points gathered

The betatesters retrieved a total of *31, 934* location points, which are divided in three categories:

- Location points with the `fused` provider: 1, 423.
- Location points with the `gps` provider: 28, 376.
- Location points with the `network` provider: 2, 135.

From this numbers the following table shows the location points gathered by user and by provider.

Providers	Anon #1	Anon #2	Anon #3	Anon #4	Anon #5	Anon #6	Anon #7	Anon #8
Fused	2	1,346	1	2	2	52	0	18
GPS	53	27,605	137	4	256	78	0	243
Network	1	520	1,029	18	164	276	0	127
Total	55	29,471	1,167	24	422	406	0	388

TABLE 5.1: Points gathered by user

Getting the numbers of the location points gathered by the betatesters it is clear that there is a real problem with the applications requesting location updates.

5.3 Staypoints running the KDD algorithm

This section presents a few examples of the execution of the KDD algorithm (section 4.2.5) in some of the datasets the betatesters gathered for performing an analysis.

All the parameters are set up in the application in the moment of executing the KDD algorithm.

The tests consider a subset of the whole set of location points from different betatesters and then, the last test, which contemplates the maximum accuracy registered, executes the KDD algorithm with the entire dataset of the betatester.

5.3.1 Tests for the dataset provided by *Anon #3*

Anon #3 provided 1167 location points in the dataset. As shown in the table 5.1, 1 are from the `fused` provider, 137 from the `gps` provider and 1029 from the `network` provider. As most of the location points are from the `network` provider, the accuracy will be poor or enough.

This is the map with the points from the dataset without performing any iteration of the KDD algorithm.

Note that, as specified in the section 4.2.3, if there is a high number of points it is performed a cluster of those points.
The tests for this dataset will be performed with the following configuration.

Values	Test 1	Test 2	Test 3	Test 4	Test 5
Dist. Threshold	2m	10m	2m	10m	100m
Time Threshold	1min	5min	1min	5min	10min
Acc. Threshold	100m	500m	1,000m	2,000m	2000m

TABLE 5.2: Test for *Anon #3*

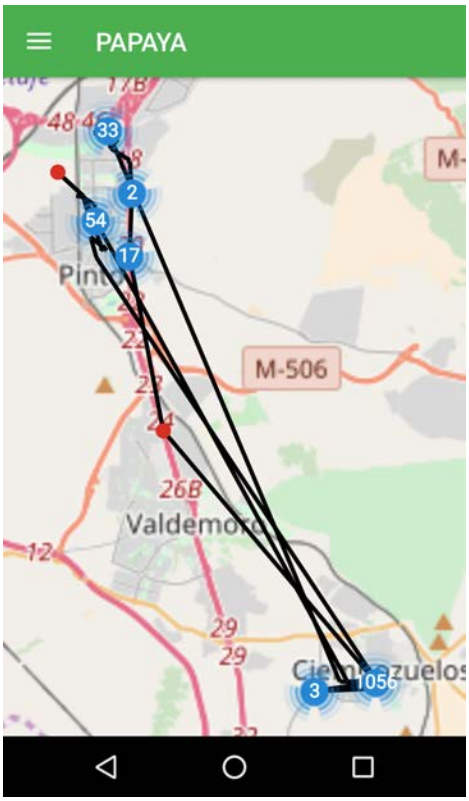

FIGURE 5.1: Location points by *Anon #3*

The tests provide the next results:

- **Test 1:** 473 staypoints.
- **Test 2:** 94 staypoints.
- **Test 3:** 504 staypoints.
- **Test 4:** 111 staypoints.
- **Test 5:** 27 staypoints.

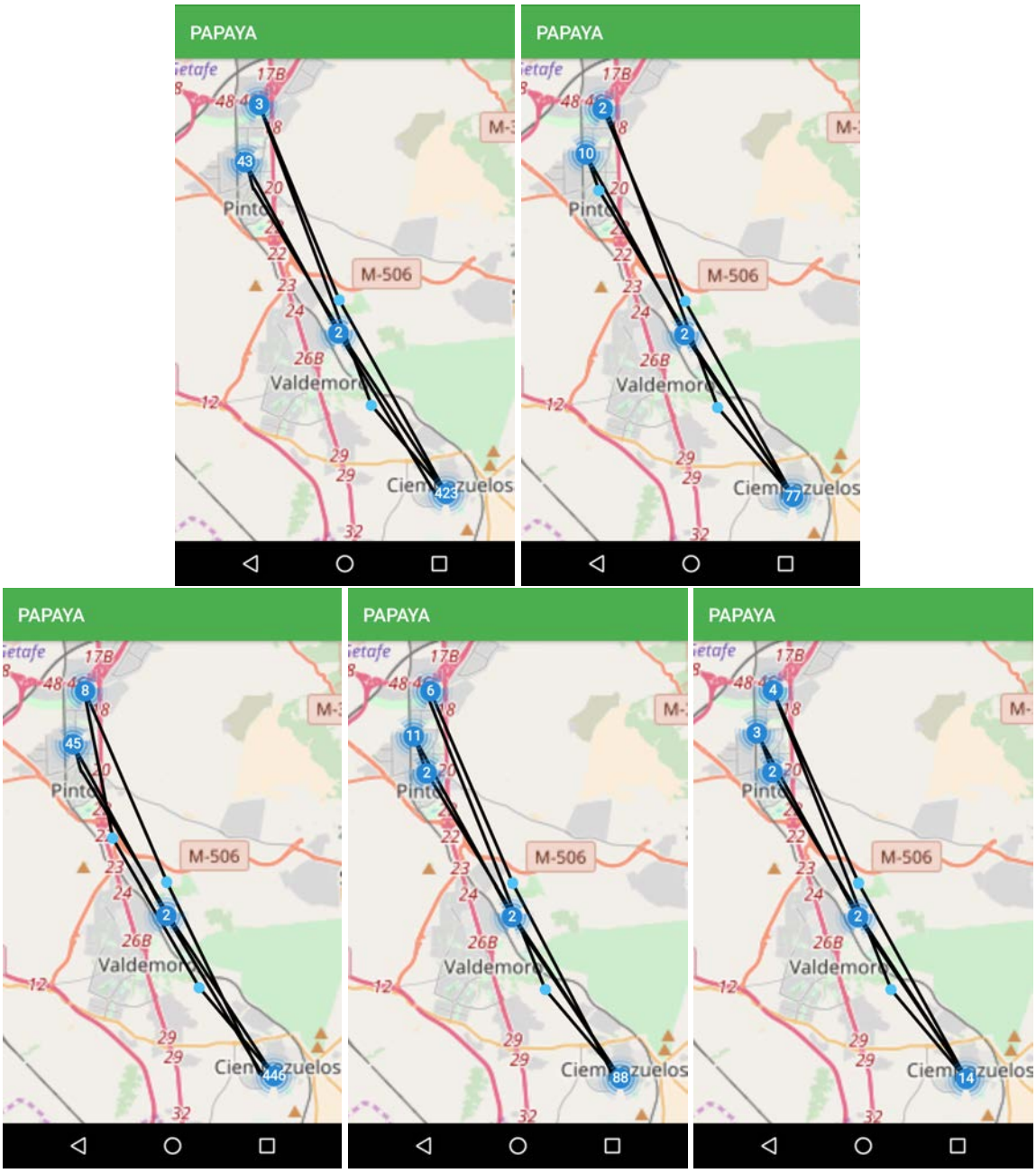

FIGURE 5.2: Staypoints calculated for *Anon #3*: Test 1, Test 2, Test 3, Test 4, Test 5

5.3.2 Tests for the dataset provided by *Anon #4*

Anon #4 provided 24 location points in the dataset. As shown in the table 5.1, 2 are from the `fused` provider, 4 from the `gps` provider and 24 from the `network` provider. As most of the location points are from the `network` provider, the accuracy will be poor or enough.

This is the map with the points from the dataset without performing any iteration of the `KDD` algorithm.

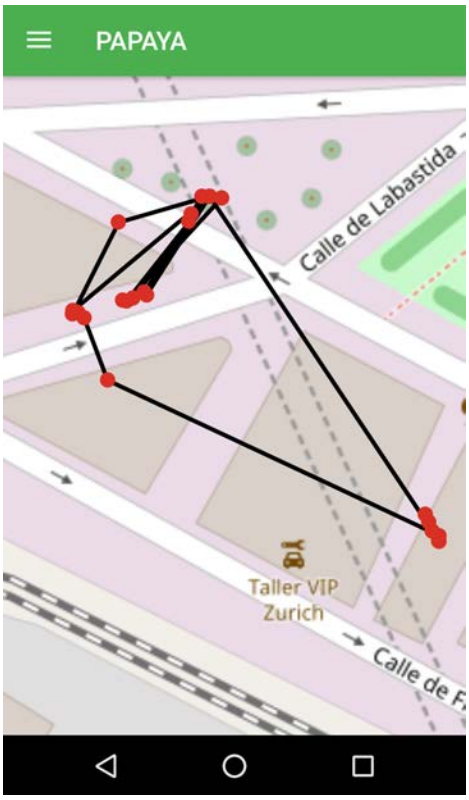

FIGURE 5.3: Location points by *Anon #4*

The tests for this dataset will be performed with the following configuration.

Values	Test 1	Test 2	Test 3	Test 4	Test 5
Dist. Threshold	2m	10m	2m	10m	100m
Time Threshold	1min	5min	1min	5min	10min
Acc. Threshold	1,000m	5,000m	50,000m	50,000m	50,000m

TABLE 5.3: Test for *Anon #4*

The tests provide the next results:

- **Test 1:** 0 staypoints.
- **Test 2:** 0 staypoints.
- **Test 3:** 15 staypoints.
- **Test 4:** 9 staypoints.
- **Test 5:** 0 staypoints.

FIGURE 5.4: Staypoints calculated for *Anon #4*: Test 3 and Test 4

5.3.3 Tests for the dataset provided by *Anon #6*

Anon #6 provided 406 location points in the dataset. As shown in the table 5.1, 52 are from the `fused` provider, 78 from the `gps` provider and 276 from the `network` provider. As most of the location points are from the `network` provider, the accuracy will be poor or enough.

This is the map with the points from the dataset without performing any iteration of the `KDD` algorithm.

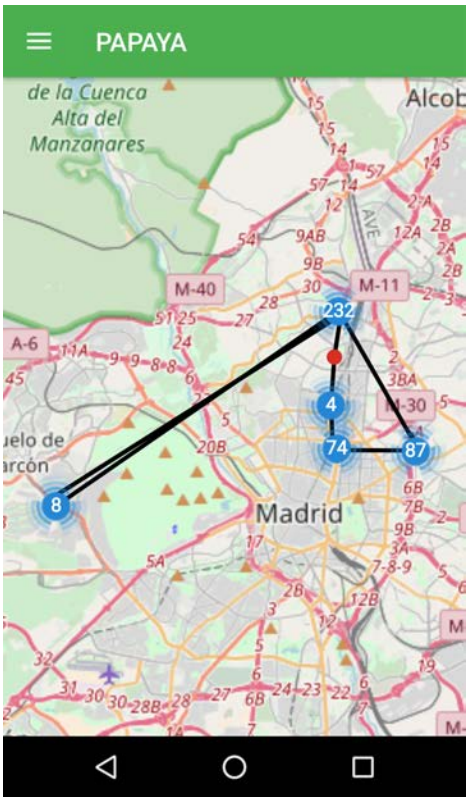

FIGURE 5.5: Location points by *Anon #6*

The tests for this dataset will be performed with the following configuration.

Values	Test 1	Test 2	Test 3	Test 4	Test 5
Dist. Threshold	2m	10m	2m	10m	100m
Time Threshold	1min	5min	1min	5min	10min
Acc. Threshold	500	1,000m	2,000m	5,000m	11,000m

TABLE 5.4: Test for *Anon #6*

The tests provide the next results:

- **Test 1:** 136 staypoints.
- **Test 2:** 33 staypoints.
- **Test 3:** 143 staypoints.
- **Test 4:** 34 staypoints.
- **Test 5:** 14 staypoints.

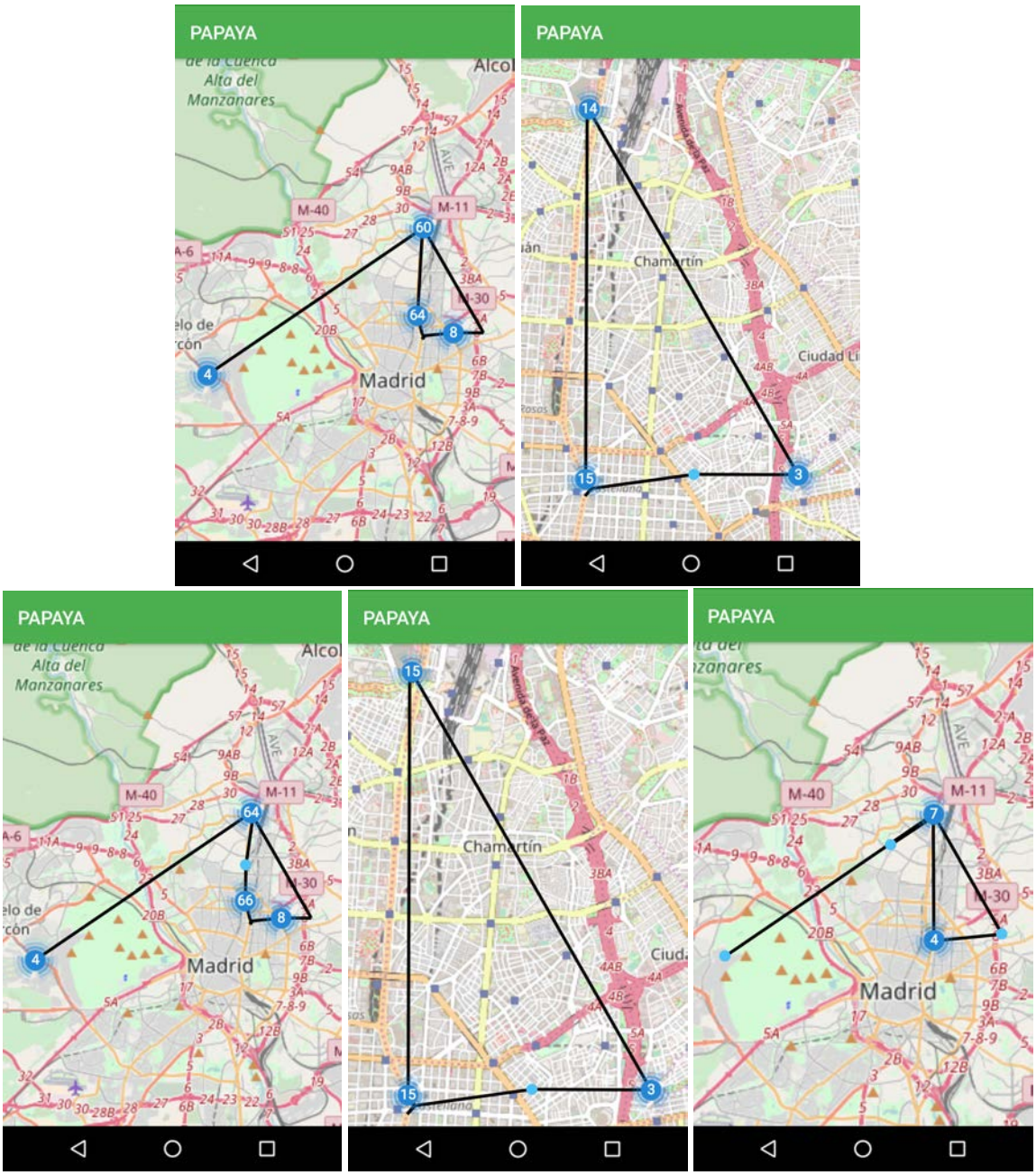

FIGURE 5.6: Staypoints calculated for *Anon #6*: Test 1, Test 2, Test 3, Test 4, Test 5

5.4 Conclusion on the testing

For the data analyzed in the section 5.3 it is possible to conclude that the staypoints represent a starting point to extracting knowledge from the user.

Taking a look at the figures 5.2, 5.4 and 5.6 it is possible to spot a pattern almost identical between all the configuration for the test. The cluster points in the tests indicate a more frequent zone than the others, meaning the user can have a personal place, like the working

place, home, or any other building or place that can provide information to infer from the user.

Also the location points come with a date. This date can indicate if the user goes to this zone in a day-to-day basis or only weekends, or certain days.

5.4.1 User feedback

This the testing performed it is not only known that the algorithm works but the user feedback is also important.

Betatesters were not really amused about the high quantity of location updates gathered by the application. Among them a 25% were really surprised that a simple application can gather that kind of information.

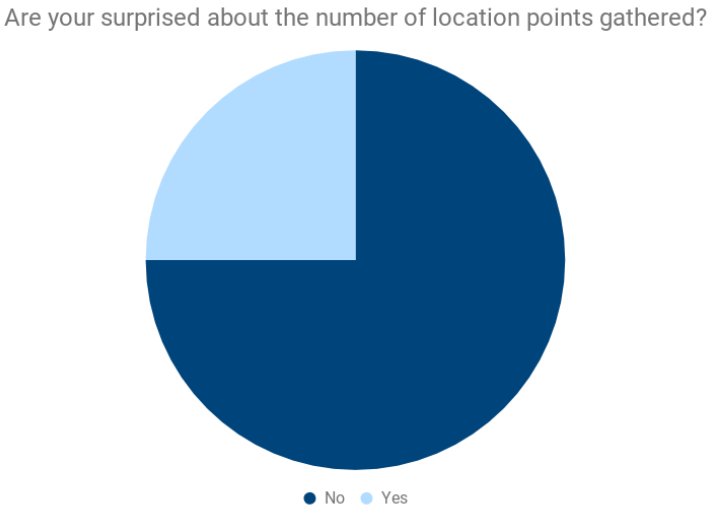

FIGURE 5.7: Responses about the number of location points

All of them coincided in that the knowledge extraction from the location data is not anything new and, knowing it is done day-to-day in the services they use, this information was already knew.

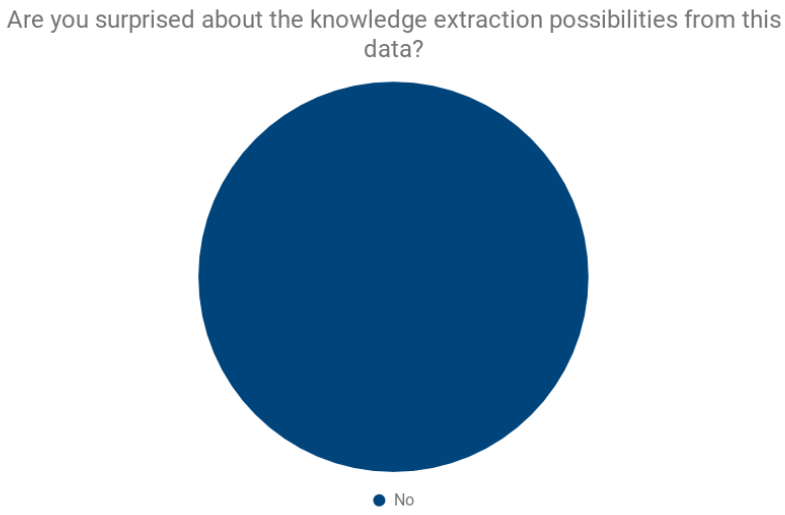

FIGURE 5.8: Responses about the knowledge extraction from the data

Although they were not amused by the quantity of the information, all of them coincide that this application is a good starting to raise awareness between the non-technical users about the high quantity of information gathered by applications. They miss a little more of context of the information, which is a point in the future work planned for this application in the chapter 8.

In terms of high performance and battery consumption they did not notice any change in the normal use of their smartphones, reflecting that the application can be used without worrying for the resources of the phone.

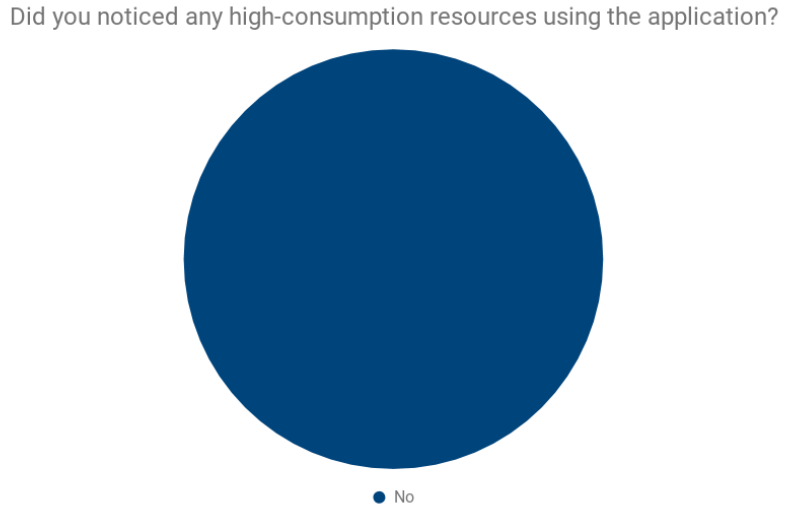

FIGURE 5.9: Responses about the resource consumption

Finally, as a personal note, they suggested features that could be developed into the application as future work (chapter 8) to improve the functionality.

6 Legal Framework and Socieconomic Context

This part of the document covers the ethical base in which the application stands for the awareness of the users, the legal and socio-economic framework and the budget invested.

Also it covers the free and open-source nature of the project and the documentation.

6.1 Ethical Consideration

Data is not sent to any third-party partner not processed outside of the smartphone. Remember that one requirement of the project is to maintain the data inside the phone.

The user, installing the application and granting all the permissions needed to execute the background service that gathers the location data and to execute the KDD algorithm, gave its consent to gather and store its location data.

For testing purposes, users who contributed to gather data to be analyzed were notified about what data will be gathered and processed by the developer of the application. They agreed to this process beforehand the application was sent to them.

6.2 Legal Framework

This section contains a picture of the laws and restrictions applied to the project and the gathered data in Android.

It is mandatory to remember that a location data, in the scope of the project, is composed by the attributes declared in the section 4.2.2 as the database scheme for the application.

Location data can be an identifiable value if it is saw as a pattern, e. g. a daily routine of an individual. Not everyone has the same daily routine and it is possible to identify someone with that [9].

6.2.1 *Protección de Datos de Carácter Personal*

This law is the predecessor of the *Protección de Datos de Carácter Personal* [31] in personal data protection.

This law considers location data as personal data (established in Title I, Article 5.1, para. *f* [32]). Also it is important to highlight that data is stored in a File (*Fichero*) to store, process, access and deletion of it (established in Title I, Article 5.1, para. *k*). Therefore this regulation must apply to this application running in European countries.

In order to make the application comply with the law it is necessary to verify this points:

- *The user must consent the application to gather location data being informed beforehand* (established in Title II, Chapter II, Article 12, 13, 14, 17 [32]).

  The application allows the user to exit from it. The consent is given when permissions are accepted and the application starts to gather data.

  If consent needs to be revoked the user can deactivate permissions to not gather data (which the application will ask again) or uninstall the application from the phone.

  If changes are done in the application and the consent is required to be given again the user will be informed beforehand accepting it.

- *Notice to the user for what purpose location data will be used for* (established in Title II, Chapter I, Article 8 [32]).

  The application informs the user about what the application is and what it does when it is opened for the first time.

- *Location services should be deactivated by default*.

  By default location is not allowed in Android. The user must allow the application to access to the location in the smartphone for the application to start running.

- *Users can revoke location permissions at any moment*.

  By using the Android permissions the application stops gathering data if location permissions are revoked.

- *Respect and satisfy the* ARCO *rights over processed data*. ARCO stands for Access (*Acceso*), Amendment and Annulment (*Rectificación y Cancelación*), Objection (*Oposición*) (established in Title III [32]).

  The user can access to their data through the application menu to visualize it in a map. Also it is possible to cancel the data erasing it from the phone in the *Settings* menu. Location data cannot be rectified because it will not be valid. If any objection is presented by the user for the data gathering, the application can be uninstalled or restricted working by permissions.

It is covered by this law the process of the data to obtain results for this document (established at Title II, Chapter I, Article 9, para. 1 [32]). To obtain this results data is given to the author by the participants in the study and it is deleted as soon as results are obtained (established in Title II, Chapter III, Article 22 [32]).

This project does not send data to any server or third-party partner, as specified in section 6.1, so the regulation about transferring data does not apply.

All the points in this sections are compliant with the application design because it is designed with the user privacy in mind. This means that, if the user does not want its data stored for a long time, it can be deleted by a simple option in *Settings* menu. The other points are related with how the information is stored and processed, and since it is all done locally, it is perfectly compliant too.

6.2.2 Foreign countries

This application may be downloaded in foreign countries so the *Protección de Datos de Carácter Personal* does not apply.

Users must verify the law of the country of residence in order to know the protection data regulation applied.

European countries must be compliant with the *General Data Protection Regulation (GDPR)* regulation.

6.2.3 Free and Open Source Software

As specified in the submission of the project to the Data Transparency Labs (DTL), the project will be published as an Open Source project.

The first step is to look for a license which suits both parts of the project, the source code and the documentation. This allows to the user to look at the source code and find if the application does as it is specified, even contributing to the project.

Source Code of PAPAYA
A suitable license for the source code of the project is the GNU GPLv3 [33], developed by the Free Software Foundation [34]. This license grants (it is shown a copy of the license note):

```
PAPAYA - A location tracker and analyzer
Copyright (C) 2018  Adolfo Santiago

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
```

```
along with this program.  If not, see
https://www.gnu.org/licensesgpl-3.0.en.html.
```

This license grants the following permissions:

- *Commercial:* The application and its derivatives may be used for commercial purposes.
- *Modification:* The application may be modified.
- *Distribution:* The application may be distributed.
- *Patents:* The license express a patent grant from contributors.
- *Private development:* The application may be modified and used in private by the own developer who modified it.

This license has the following limitations:

- *Warranty:* This license includes a limitation of warranty.
- *Liability:* This license includes a limitation of liability.

This license has the following conditions:

- *Distribution of the source code:* The source code may be distributed. If so a copy of the license must be attached. It is allowed to distribute verbatim copies.
- *Modified version:* If a copy of the modified version is distributed, it must be under the same license of the original code plus with all the copyright and license notices of previous developers. If the modified source code is going to be released under another license, it must be a `GNU GPLv3`-compatible license.
- *Copyright notice:* The source code must have a notice of the license attached into the source file plus a copy of the full text of the license.

The `GNU GPLv3` license is compatible with the `Apache License v2`, the `GNU Lesser General Public License v3.0` license [35] and the `MIT` license [36]. Those licenses are for the third-party libraries referred in the section 3.2.2 and are compatible with the chosen license for this project in terms of source code.

Documentation of PAPAYA

The documentation of the project (this document and all the documentation generated after) will be available under the Creative Commons 3.0 (`CC BY-NC-ND 3.0`) [37], which makes all the documentation shareable under this conditions:

- *Attribution:* You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- *Non-Commercial:* You may not use the material for commercial purposes.

- *No-Derivatives:* If you remix, transform, or build upon the material, you may not distribute the modified material.

6.3 Socioeconomic Context

This section contains the budget of the project estimated for a developer plus some equipment and software used in it.

6.3.1 Budget

The budget of the project is divided in two categories:

- Direct costs: This category contains the costs of software, equipment and materials along of the personnel costs.
- Indirect costs: Indirect costs go beyond the expenses associated with creating a particular product to include the price of maintaining the entire company. For this project it was decided to fix a 15% of the direct costs.

This project was made by a single developer assuming different roles.

Personnel	Salary (€/hour)	Hours	Cost
Project Manager	30	250	7,500.00 €
Developer	25	250	6,250.00 €
Q&A Engineer	20	130	2,600.00 €
Total			**16,350.00 €**

TABLE 6.1: Personnel total costs

It is necessary to add the imputable cost of the equipment used in the process of developing the project, including the infrastructure needed.

Equipment	Total Price	Life Span	Imputable Cost
MacBook Pro (Retina 13", 2015)	1,800.00 €	48 months	180.00 €
LG Nexus 4	350.00 €	10 months	35.00 €
LG Nexus 5	350.00 €	10 months	35.00 €
GitHub Developer Account	7 €/month	10 months	70.00 €
Total			**320.00 €**

TABLE 6.2: Equipment costs

The next table represents the total amount of the previous computed costs, to which the implicit costs are added (15% of the direct costs).

Concept	Total
Personnel Costs	16,350.00 €
Equipment Costs	320.00 €
Implicit Costs	2,452.50 €
Total	**19,122.5 €**

TABLE 6.3: Project costs

To get the final amount of the budget it is necessary to add a 10% of the sum at 6.3 quantity representing both risk, that is, additional money used for unexpected expenses. The final cost of the project is **21,034.75 €**.

7 Conclusions

The study of user trajectory is still in its early stages. While there are some work performed with users in the real world the studies are mostly performed by using social networks with geolocation or GPS specialized devices.

The main goal of this thesis is to provide a way to start raising a level of awareness of privacy between the users. First the project was implemented and tested and then given to users to use as a day-to-day application. Then an analysis is performed to extract some knowledge from the data, in this project, the staypoints of the location data in datasets.

The testing shown that the location updates in a phone are a high problem, giving that most of the application which request location updates are those where the main content is not related with location. The extraction of staypoints with the KDD algorithm shows that the original data is related to the most frequent places as a pattern of movements. From that set of points, original or the staypoints, is possible to extract knowledge and real data like directions.

The main goal of the application has been completed and the basic application is working in the real world. Making the research in this thesis proves that the data is reliable and the awareness of the privacy is improved when data can be visualized and analyzed locally, without leaving the phone, meaning is user-privacy first.

The data has to be comprehensible too. The visualization on a map is a way to show the user the gathered data to an unexperienced user and helps to an experienced user to visualize the data. Even the application is in its early stage it is a good start to improve the privacy level on the Android smartphones.

With this project it is possible to make academic research from location updates and from location updates to extract knowledge from the users while preserving the user privacy with their data. Also it is the start as a researcher and learning to apply the scientific method to learn traveling from a hypothesis to a conclusion analyzing the data with the results and finding out if the premises are valid.

As a personal note from the author of the thesis all the knowledge acquired in the university and to do a basic research work contributing to the academic world. Also this project means a lot as a privacy-wise person and shows that users really have to worry about their privacy.

8 Future work

This project has a good potential in giving the users a good and wide visualization of which data is gathered by all the applications installed in their phone. It is far from done because it lacks functionality it is planned to develop and integrate into the application.

One of the "problems" for this project was the security of Android. The API provided by the operating system lacks giving the data the project needed in order to get all the necessary information. For this case it is planned to add a Virtual Private Network (VPN) in form of Man-In-The-Middle (MITM). This setup will allow to snoop into the traffic in real-time to know if location data is leaked and by which application. The analyzed data will remain locally analyzed.

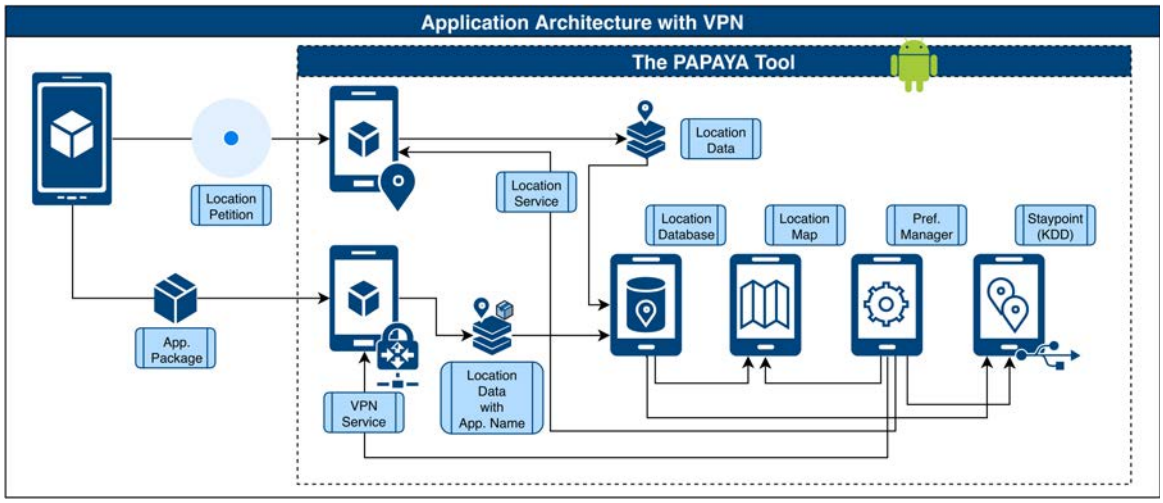

FIGURE 8.1: New Application Architecture with VPN

Another future step towards improving the application is to enhance the data visualization by making a difference between applications in the map, so the users can see specifically the location data and by which application. There is also a necessity to improve the data visualization to handle big datasets of points for medium range phones. Another feature will be to tag those locations in the map with the time-span mark, so the user can see if a location update (or, generally, location updates), were passive or not expected to occur.

The application will be updated to get enhanced by the new features the Android platform offers in futures updates and improving the interface to the Material Design guidelines. This allows to keep up-to-date with the changes in battery consumption and general performance, making the application battery-wise keeping all the features implemented. Also the third-party libraries used in the project will be updated to the latest version to use new features such a better handling of high sized datasets or improved caching back-end.

Finally the application will be available for the users through the Play Store [38] and F-Droid [39]. This allow the users to install the application and have a tool for awareness of their privacy. It is important that users get concerned about their data, because nowadays it is an important portion of all the data companies gather about users. Also the source code and the documentation will be published at GitHub for making it available to the general public.

Acronyms

2FA 2-Factor Authentication. 1

A-GPS Assisted GPS. 6, 19

ads Advertisement. viii, 1

API Application Proggramming Interface. 7, 15, 46

GPS Global Positioning System. 5, 6, 19, 25, 45

GUI Graphical User Interface. 13, 18, 49

ISP Internet Service Provider. 1

KDD Knowledge Discovery in Databases. vi, viii, ix, 2, 9, 22–26, 28, 30, 32, 34, 39, 45

MVC Model-View-Controller. 17, 18

PAPAYA **P**rivacy leackages from **a**p**p** tr**a**jectory d**a**ta. viii, ix, 2, 6, 17, 41, 42

PP Privacy Policy. 1, 2

SMS Short Message Service. 1

USB Universal Serial Bus. 9, 24

Glossary

Android Operating system based on Linux and other open source tools available for touchscreen devices. 1, 3–8, 15, 18–22, 24, 29, 39, 40, 45, 46, 49

Android Debug Bridge (ADB) Android Debug Bridge (ADB) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device [40]. 7

Creative Commons Creative Commons is a global nonprofit organization that enables sharing and reuse of creativity and knowledge through the provision of free legal tools. 42

Data Transparency Labs (DTL) Data Transparency Labs is an interconstitutional collaboration seeking to create a global community working to advance online personal data transparency. 3, 4, 41

Free Software Foundation The Free Software Foundation (FSF) [34] is a nonprofit with a worldwide mission to promote computer user freedom. They defend the rights of all software users. 41, 49

General Data Protection Regulation (GDPR) The General Data Protection Regulation (GDPR) is a european law approved in April 27, 2016, to improve the law over the personal data of the users and how it must be shared, procesed, used and deleted between countries and companies. 41

GNU GPLv3 The GNU General Public License v3 (GNU GPLv3) is a free software license developed by the Free Software Foundation. 41

Google Play Store The Google Play Store is a free and paid application market where users can download applications for the Android operating system. 3

Integrated Development Environment (IDE) An Integrated Development Environment (IDE) is a set of tools (editor, compiler, debugger...) for software development accessed through a single GUI. 7

Java Virtual Machine (JMV) The Java Virtual Machine (JVM) provide an runtime environment to execute Java applications compiled to the Java bytecode. 7

Man-In-The-Middle (MITM) A MITM (attack) is an attack that intercepts communication between two systems [41]. 46

Open Source Open Source is a term coined to define a project released with certain conditions defined at *The Open Source definition* [42] and under one of the Open Source licensed listed at *Open Source Licenses by Category* [43]. 41

Protección de Datos de Carácter Personal The spanish law in personal data protection complying with the General Data Protection Regulation (GDPR). 39, 41

root *root* is the process of being an administrator of the device and having permissions to access to the internal storage and modifying internal files, and giving this access to third party applications. 3, 5, 8, 18

staypoint A *staypoint* is a location point where an user stayed for a fixed time span. 5, 6

Version Control System (VCS) Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. 7

Virtual Private Network (VPN) VPN. 46

References

[1] (Apr. 19, 2018). Facebook's data policy, Facebook, [Online]. Available: https://www.facebook.com/privacy/explanation (visited on 05/26/2018).

[2] (May 25, 2018). Twitter's privacy policy, [Online]. Available: https://twitter.com/content/twitter-com/legal/en/privacy.html (visited on 05/26/2018).

[3] J. Valentino-DeVries, "Service meant to monitor inmates' calls could track you, too", *The New York Times*, May 10, 2018, ISSN: 0362-4331. [Online]. Available: https://www.nytimes.com/2018/05/10/technology/cellphone-tracking-law-enforcement.html (visited on 05/26/2018).

[4] (2017). Home, Data Transparency Lab, [Online]. Available: http://datatransparencylab.org/ (visited on 05/26/2018).

[5] (2018). Traccar, Open Source GPS Tracking Software, [Online]. Available: https://www.traccar.org/ (visited on 05/29/2018).

[6] (2018). ICSI haystack project, [Online]. Available: https://www.haystack.mobi/ (visited on 06/13/2018).

[7] R. B. Ribas Manero, "Privacy implications of android applications in the presence of third-party libraries", 2014.

[8] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones", p. 15, Oct. 3, 2010.

[9] I. Liccardi, A. Abdul-Rahman, and M. Chen, "I know where you live: Inferring details of people's lives by visualizing publicly shared location data", ACM Press, 2016, pp. 1–12, ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858272. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2858036.2858272 (visited on 06/10/2018).

[10] M. Spreitzenbarth, S. Schmitt, and F. Freiling, "Comparing sources of location data from android smartphones", in *Advances in Digital Forensics VIII*, G. Peterson and S. Shenoi, Eds., vol. 383, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 143–157, ISBN: 978-3-642-33961-5 978-3-642-33962-2. DOI: 10.1007/978-3-642-33962-2_10. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33962-2_10 (visited on 06/13/2018).

[11] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining individual life pattern based on location history", IEEE, 2009, pp. 1–10, ISBN: 978-1-4244-4153-2. DOI: 10.1109/MDM.2009.11. [Online]. Available: http://ieeexplore.ieee.org/document/5088915/ (visited on 06/13/2018).

[12] (2018). Kotlin programming language, Kotlin, [Online]. Available: https://kotlinlang.org/ (visited on 06/08/2018).

[13] (Feb. 18, 2018). Home, OSMDroid, [Online]. Available: https://osmdroid.github.io/osmdroid/ (visited on 06/11/2018).

[14] (2018). Website, OpenStreetMap, [Online]. Available: https://www.openstreetmap.org/ (visited on 06/11/2018).

[15] MKer. (Jun. 11, 2018). Home, osmbonuspack: A third-party library of (very) useful additional objects for osmdroid, [Online]. Available: https://github.com/MKergall/osmbonuspack (visited on 06/11/2018).

[16] (Jan. 4, 2018). SQLiteImporterExporter: A light weight library for exporting and importing sqlite database in android, GitHub, [Online]. Available: https://github.com/androidmads/SQLiteImporterExporter (visited on 06/11/2018).

[17] (2018). Git, [Online]. Available: https://git-scm.com/ (visited on 06/08/2018).

[18] (2018). Mercurial SCM, [Online]. Available: https://www.mercurial-scm.org/ (visited on 06/08/2018).

[19] (2018). Apache subversion, [Online]. Available: https://subversion.apache.org/ (visited on 06/08/2018).

[20] (2018). GitHub, [Online]. Available: https://github.com/ (visited on 06/08/2018).

[21] (2018). Home, Trello, [Online]. Available: https://trello.com/ (visited on 06/08/2018).

[22] Atlassian. (2018). Website, Kanban - A brief introduction, [Online]. Available: https://www.atlassian.com/agile/kanban (visited on 06/08/2018).

[23] (Jun. 6, 2018). LocationManager, Android Developers, [Online]. Available: https://developer.android.com/reference/android/location/LocationManager (visited on 06/10/2018).

[24] (Jun. 6, 2018). Service, Android Developers, [Online]. Available: https://developer.android.com/reference/android/app/Service (visited on 06/11/2018).

[25] (Mar. 21, 2018). FusedLocationProviderClient, Google Developers, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient (visited on 06/13/2018).

[26] G. M. Djuknic and R. E. Richton, "Geolocation and assisted GPS", p. 3, 2001.

[27] (2018). SQLite home page, [Online]. Available: https://sqlite.org/index.html (visited on 06/11/2018).

[28] (Jun. 6, 2018). Fragments, Android Developers, [Online]. Available: https://developer.android.com/guide/components/fragments (visited on 06/18/2018).

[29] (Jun. 6, 2018). BroadcastReceiver, Android Developers, [Online]. Available: https://developer.android.com/reference/android/content/BroadcastReceiver (visited on 06/12/2018).

[30] F. Ivis, "Calculating geographic distance: Concepts and methods", p. 10, 2006.

[31] "Ley orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal", p. 25, Dec. 14, 1999. [Online]. Available: https://boe.es/buscar/pdf/1999/BOE-A-1999-23750-consolidado.pdf.

[32] *Protección de datos de caracter personal*. Madrid: Agencia Estatal Boletín Oficial del Estado, 2014, OCLC: 908337033, ISBN: 978-84-340-2157-0.

[33] (Mar. 13, 2018). Home, The GNU General Public License v3, [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html (visited on 06/01/2018).

[34] (2018). Front page, Free Software Foundation, [Online]. Available: https://www.fsf.org/ (visited on 06/01/2018).

[35] (2018). Home, GNU Lesser General Public License v3.0, [Online]. Available: https://www.gnu.org/licenses/lgpl-3.0.en.html (visited on 06/11/2018).

[36] (2018). Open source initiative, The MIT License, [Online]. Available: https://opensource.org/licenses/MIT (visited on 06/11/2018).

[37] (2018). Creative commons, Attribution-NonCommercial-NoDerivs 3.0 Unported - CC BY-NC-ND 3.0, [Online]. Available: https://creativecommons.org/licenses/by-nc-nd/3.0/ (visited on 06/01/2018).

[38] (2018). Home, Google Play Store, [Online]. Available: https://play.google.com/store (visited on 06/07/2018).

[39] (Jun. 5, 2018). Home, F-Droid - Free and Open Source Android App Repository, [Online]. Available: https://f-droid.org/ (visited on 06/07/2018).

[40] (Jun. 5, 2018). Android debug bridge (adb), Android Developers, [Online]. Available: https://developer.android.com/studio/command-line/adb (visited on 06/08/2018).

[41] (Aug. 31, 2015). Man-in-the-middle attack - OWASP, OWASP Wiki, [Online]. Available: https://www.owasp.org/index.php/Man-in-the-middle_attack (visited on 06/11/2018).

[42] (Mar. 22, 2007). Open source initiative, The Open Source Definition, [Online]. Available: https://opensource.org/osd (visited on 06/01/2018).

[43] (2018). Open source initiative, Open Source Licenses by Category, [Online]. Available: https://opensource.org/licenses/category (visited on 06/01/2018).